SIAM J. COMPUT.
Vol. 36, No. 6, pp. 1729–1747

PHYSICAL LIMITS OF HEAT-BATH ALGORITHMIC COOLING*

LEONARD J. SCHULMAN†, TAL MOR‡, AND YOSSI WEINSTEIN‡

Abstract. Simultaneous near-certain preparation of qubits (quantum bits) in their ground states is a key hurdle in quantum computing proposals as varied as liquid-state NMR and ion traps. "Closed-system" cooling mechanisms are of limited applicability due to the need for a continual supply of ancillas for fault tolerance and to the high initial temperatures of some systems. "Open-system" mechanisms are therefore required. We describe a new, efficient initialization procedure for such open systems. With this procedure, an n-qubit device that is originally maximally mixed, but is in contact with a heat bath of bias $\varepsilon \gg 2^{-n}$, can be almost perfectly initialized. This performance is optimal due to a newly discovered threshold effect: For bias $\varepsilon \ll 2^{-n}$ no cooling procedure can, even in principle (running indefinitely without any decoherence), significantly initialize even a single qubit.

Key words. quantum computation, state preparation, thermodynamics

AMS subject classifications. 68W01, 80A99

DOI. 10.1137/050666023

1. Introduction.

Quantum computation poses a difficult experimental challenge. Simultaneous near-certain preparation of qubits (quantum bits) in their ground states is a key hurdle in proposals as varied as NMR and ion traps [8, 19, 9, 13, 10, 11]. Such "cooling" (also known as "biasing" or "polarizing") is required both for initiation of the computation [2] and in order to supply ancillas for fault tolerance as the computation proceeds.

Cooling of quantum systems has long been essential in a variety of experimental contexts unrelated to quantum computation, and is performed by processes that directly cool the system such as laser cooling in ion traps or application of strong magnetic fields in NMR. Spin exchange has also been employed in order to transfer highly cooled states into the desired system from another that is more readily directly cooled [4, 14, 24]. In all these methods, the temperature is limited by the original cooling process.

Algorithmic cooling. It is in principle possible, however, to reach even lower temperatures, by application of certain logic gates among the qubits [22]. (Even prior to quantum computation the need for signal amplification in NMR imaging led to the implementation of a basic 3-qubit logic gate [23].) In several quantum computation proposals this kind of improvement in cooling is necessary due to the requirement that a large number of qubits all be, with high probability, simultaneously in their ground states.

We distinguish between closed- and open-system algorithmic cooling methods. In the former [22] an initial phase of physical cooling is performed which reduces the

*Received by the editors March 9, 2005; accepted for publication (in revised form) October 6, 2006; published electronically March 19, 2007.
http://www.siam.org/journals/sicomp/36-6/66602.html

†California Institute of Technology, MC 256-80, Pasadena, CA 91125 (schulman@caltech.edu). The work of this author was supported in part by the NSF (PHY-0456720 and CCF-0524828), the ARO (W911NF-05-1-0294), the Mathematical Sciences Research Institute, and the Okawa Foundation.

‡Technion - Israel Institute of Technology, Haifa 32000, Israel (talmo@cs.technion.ac.il, yossiv@cs.technion.ac.il). The work of these authors was supported in part by the Israel Ministry of Defense and by the Institute for Future Defense Research at the Technion.

entropy of the system. Then in the closed phase an entropy preserving (unitary) algorithmic process is performed on the qubits. By contrast in an open process [5] some of the qubits of the system can be cooled by external interaction even during (or at interruptions in) the quantum computation. Open-system cooling places an additional experimental difficulty: Computation qubits must not decohere during the process of cooling other qubits which, at another stage, they must interact with. Nonetheless closed-system cooling appears to be insufficient for two reasons. The first applies specifically to liquid-state NMR quantum computing, where the initial entropy-reducing preparation is quite weak: the probability of the ground state of each qubit exceeds the probability of the excited state by the small factor of $e^{2\varepsilon} \approx 1+10^{-5}$. In the subsequent closed phase an ε^2 fraction of the qubits can be prepared in highly cooled states [22] (and see [23, 7] for experimental demonstrations of key steps); for information-theoretic reasons this fraction is best possible, but at the current value of ε it is too small for effective implementation of a quantum computer. The second reason applies more broadly. Any quantum computing implementation must cope with noise. Fault-tolerance mechanisms have been designed that can do so [1], if the noise level is below a specified threshold (estimated to be between 10^{-4} and 10^{-2} per qubit per operation [16]) and if a continual supply of "ancillas" (qubits which are initialized in a known state) is available. Ancilla initialization need not be perfect, but the error cannot exceed the same fault-tolerance threshold. In ion traps, for example, direct cooling can place qubits in their ground states with probability ≈ 0.95, a level that necessitates further cooling to exceed the threshold [15, 3]. Since fresh ancillas are needed in each time step, either a very large supply must be chilled in advance and maintained without substantial decoherence, or—more likely—an open-system approach must be adopted in which registers are cooled on a regular basis.

It is necessary, therefore, to study effective means for open-system algorithmic cooling. A suggested framework (called the "heat-bath" approach) was made in [5]. A heat-bath device comprises two types of qubits—some that are hard to cool (but relax slowly) and others that are readily cooled (but relax rapidly). The former are computation qubits and the latter are "refrigerants." At chosen times, the computation and refrigerant qubits can undergo joint unitary interaction (such as spin exchange). A similar framework is contemplated for ion trap quantum computers [3]—the computation ions are not cooled directly, due to the decoherence that this causes; instead they are cooled by interaction with separate refrigerant ions that have been directly laser-cooled.

Results. In this paper we establish the theoretical limits for cooling on heat-bath devices. We introduce a cooling mechanism achieving much higher bias amplification than given previously. We explicitly bound the number of cooling steps required in our amplification process, a crucial matter, since any cooling process must be carried out within the relaxation times of the computation qubits. Finally, we show that our method is optimal in terms of entropy extraction per cooling step. In the course of doing so we discover a threshold phenomenon: significant initialization cannot be achieved at all unless ε, the bias that can be imparted to the rapidly relaxing qubits, is asymptotically above 2^{-n}. The proof uses majorization inequalities to convert the problem to analysis of a certain combinatorial "chip game."

For specificity we assume that the quantum computer has $n-1$ computation qubits, and an nth refrigerant qubit that is in contact with the heat bath. The cooling step, ι, has the effect of changing the traced density matrix of the nth qubit

to

$$\rho_\varepsilon = \frac{1}{e^\varepsilon + e^{-\varepsilon}} \begin{pmatrix} e^\varepsilon & 0 \\ 0 & e^{-\varepsilon} \end{pmatrix} \tag{1.1}$$

(no matter what the previous state was). In between cooling steps, reversible (unitary) quantum logic gates can be applied to the register of n qubits. Let $\mathcal{I}_n$ be the density matrix of the maximally mixed state over the 2^n-dimensional Hilbert space ($\mathcal{I}_n = 2^{-n}\times$ the identity matrix of dimension 2^n). The question is, Starting from $\mathcal{I}_n$, and using these operations, how different from $\mathcal{I}_n$ can we make the density matrix of the device?

There is little a priori reason to expect any limit on the difference. To speak (imprecisely) in terms of temperature, we have already pointed out that the temperature of the heat bath is not a lower bound on the achievable temperature of the device, because we can use logic gates and energy to implement a heat engine (refrigerator). This being so, there is no natural lower bound on the achievable temperature short of absolute zero. It is therefore fascinating that a positive lower bound exists. The bound derives not from entropic considerations but from finite-size effects. The precise statement is not in terms of temperature but in terms of the maximum probability of any state. (For a Gibbs distribution this would be a ground state.)

THEOREM 1.1 (physical limit). *No heat-bath method can increase the probability (i.e., $|amplitude|^2$) of any basis state from its initial value, 2^{-n}, to any more than* $\min\{2^{-n}e^{\varepsilon 2^{n-1}}, 1\}$. *This conclusion holds even under the idealization that an unbounded number of cooling and logic steps can be applied without error or decoherence.*

This shows that if $\varepsilon \ll 2^{-n}$, then the variation distance between the uniform distribution, and any distribution reachable by cooling, is $\ll 1$.

On the flip side, it was shown in [12] how to produce (at small ε) a qubit of bias $(3/2)^{(n-2)/2}\varepsilon$. We improve on this result and establish a converse to Theorem 1.1, using a specific cooling procedure, the PPA, described below. For convenience let $\tilde{\varepsilon} = \tanh\varepsilon$. (For small ε, $\tilde{\varepsilon} \approx \varepsilon$.) We present the converse in two slightly incomparable forms.

THEOREM 1.2 (threshold effect). *If $\tilde{\varepsilon} \geq 2^{4-n}$, the PPA increases the variation distance from uniform to $\Theta(1)$. This occurs within $\tilde{\varepsilon}^{-2}$ cooling steps.*

THEOREM 1.3 (cold qubit extraction). *Within $4n\tilde{\varepsilon}^{-2}(1+\log(1/\tilde{\varepsilon}))$ cooling steps, the PPA creates a probability distribution in which with probability at least $1-O(\frac{1}{1+\log 1/\tilde{\varepsilon}})$, all of the first $n-(1+o(1))\log_2 1/\tilde{\varepsilon}$ bits are $|0\rangle$'s (where $o(1)$ denotes a term tending to 0 as $\tilde{\varepsilon}$ tends to 0).*

This extraction procedure is useful for quantum computing (it extracts qubits of bias almost 1, i.e., that are almost certainly in their ground state) so long as ε is above $n2^{-n}$.

The notion that the computation qubits are entirely insulated from the environment is of course merely a simplification good for moderate time spans. To be useful, algorithms must converge to the desired state within the relaxation time of the computation qubits. Next we show that the PPA is near-optimal in terms of the number of cooling steps.

THEOREM 1.4 (cooling steps required). *Any algorithm which creates a bit of constant bias requires $\Omega(\tilde{\varepsilon}^{-2})$ cooling steps.*

Finally, since the computations in the PPA vary in a complex way depending upon the value of n, we accompany the above results with another simpler cooling

procedure that applies transpositions and reversible 3-qubit majorities in a recursive pattern, and performs fairly effective cooling. This procedure is a slight modification, to achieve better asymptotics, of one given in [12]. Let $\phi = (1+\sqrt{5})/2$, let F_k be the kth Fibonacci number, and let $N = \min\{n, \lceil \log_\phi 1/\tilde{\varepsilon} \rceil\}$; the cooling algorithm $\mathcal{F}$ mentioned in the theorem is described in section 8.

THEOREM 1.5 (simple cooling algorithm). *The cooling algorithm $\mathcal{F}$ is NC1-uniform and, when run on an N-bit device, creates a bit of bias $\Omega(\tilde{\varepsilon} F_N)$ within runtime (counting both cooling steps and logic gates)* $\exp(O(N \log N))$.

Comment. The reader will have noticed that while we speak of "cooling," the algorithms are characterized not in terms of the final temperatures achieved in the qubits but in terms of other desirable properties of the final probability distributions. There are two reasons for this. The first is that other properties, especially as in Theorem 1.3, are more germane to the application to quantum computing. The second is that unambiguous assignment of a temperature to a probability distribution depends on the latter being a Gibbs distribution for some Hamiltonian describing the system; but the distributions produced by algorithmic cooling need not be Gibbs distributions. In particular, the PPA does not produce a Gibbs distribution.

Other applications of algorithmic cooling. A central point of this paper is the firm limit that Theorem 1.1 sets on the cooling parameter ε in order that the heat-bath method be useful for quantum computation. However, it is important to note that heat-bath cooling algorithms (the PPA or others) may be viable for other applications even at smaller ε. Specifically, algorithmic cooling is likely to find significant application in the scientific and medical imaging applications for which NMR technology is already in wide use. The signal-to-noise ratio in NMR imaging is proportional to the polarization of the nuclear spins and to the square root of the duration of the scan; since the duration is often limited in medicine by the need to immobilize the patient, improved sensitivity demands increased polarization. In other applications the benefit of increased polarization is in decreased scan times. Algorithmic cooling of a few nuclear spins may therefore be highly beneficial even in the range $\varepsilon \ll 2^{-n}$ that is not adequate for quantum computation. For example, perfect implementation of the PPA on a 5-qubit molecule (four computation qubits and one refrigerant) would yield a qubit of bias 8ε, implying a 64-fold decrease in scan duration compared to cooling without algorithmic amplification.

An abridged version of this paper appeared in [21].

2. Reduction of quantum to classical cooling. In preparation for the proofs of Theorems 1.1–1.5 we start with a reduction that significantly simplifies the rest of our task. Recall that the heat-bath quantum computer is assumed to start in the maximally mixed density matrix, $\mathcal{I}_n$. Any cooling step ι changes the traced density matrix on the nth qubit to the matrix given in (1.1). To see how this affects the entire density matrix, suppose that before the cooling step, the quantum computer is in a $2^n \times 2^n$ density matrix

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{12}^\dagger & M_{22} \end{pmatrix}, \tag{2.1}$$

where the states $|0\rangle$ and $|1\rangle$ of the nth qubit partition the density matrix into these four parts. Application of ι effects the following transformation:

$$M \xrightarrow{\iota} \rho_\varepsilon \otimes (M_{11} + M_{22}) = \frac{1}{e^\varepsilon + e^{-\varepsilon}} \begin{pmatrix} e^\varepsilon (M_{11} + M_{22}) & 0 \\ 0 & e^{-\varepsilon}(M_{11} + M_{22}) \end{pmatrix}. \tag{2.2}$$

Between cooling steps, quantum logic gates can be applied to the system. These act on the density matrix as conjugations by unitary operators. If there are $r+1$ cooling steps, let these unitaries be $u_1, \ldots, u_r$. These unitary operators are constrained to be implementable by local quantum logic gates; for the limit on achievable cooling (Theorem 1.1), we may ignore this constraint and allow the unitaries to be arbitrary. For unitary u let $\overline{u}$ denote the corresponding conjugation operator.

The eigenvalues of a density matrix are the probabilities with which the spectral basis states are measured; by an inequality of Schur, the spectral basis gives measurement probabilities that are furthest from uniform, in the sense of majorization (see [18, section 9B]). A probability vector $p = (p_1, \ldots)$ is said to majorize another $p' = (p'_1, \ldots)$ (written $p \succeq p'$) if there exists a doubly stochastic matrix D such that $(p_1, \ldots)D = (p'_1, \ldots)$. This is a partial (pre-)order on probability distributions in which the singular distribution $(1, 0, 0, \ldots)$ dominates all others, while the uniform distribution is dominated by all. Schur's inequality is that the eigenvalues of a Hermitian matrix majorize its diagonal entries. A density matrix h is said to majorize another h' (written $h \succeq h'$) if the eigenvalues of h majorize those of h'.

Domination in majorization implies domination in any of the other measures we are interested in, such as variation distance from uniform, or the sum of the largest K probabilities (for a fixed K). So our concern is the following: If $\overline{u_1}, \ldots, \overline{u_r}$ represent the reversible actions of an algorithm between its cooling steps (each acting on the density matrix as conjugation by a unitary operator), how different can the eigenvalues of $\iota\, \overline{u_r}\, \iota\, \cdots\, \overline{u_1}\, \iota\, \mathcal{I}_n$ be from those of $\mathcal{I}_n$ (in which all equal 2^{-n})?

A *classical* cooling algorithm is one that uses only reversible (deterministic) classical logic gates between cooling steps. In this case each operator $\overline{u_i}$ acts on the density matrix as conjugation by a permutation matrix. Observe that a $2^n \times 2^n$ diagonal density matrix represents a probability distribution over the basis states $|0 \ldots 0\rangle, \ldots, |1 \ldots 1\rangle$.

PROPOSITION 2.1 (classical cooling). *Let h be a $2^n \times 2^n$ diagonal density matrix. Given any quantum logic steps $\overline{u_1}, \ldots, \overline{u_r}$, there are classical steps $\overline{\pi_1}, \ldots, \overline{\pi_r}$ such that $\iota\, \overline{\pi_r}\, \iota\, \cdots\, \overline{\pi_1}\, \iota\, h$ majorizes $\iota\, \overline{u_r}\, \iota\, \cdots\, \overline{u_1}\, \iota\, h$.*

For a density matrix M with eigenvectors $v_1, \ldots, v_{2^n}$ listed in decreasing order of their eigenvalues $\lambda_1 \geq \cdots \geq \lambda_{2^n}$, let w be a unitary operator which arranges the eigenvectors so that they correspond, in order, to the vectors $|0..00\rangle, |0..01\rangle, |0..10\rangle, \ldots, |1..11\rangle$ (recall that the "cooling bit" that is in contact with the reservoir is the nth or rightmost bit). Then, acting on M with $\overline{w}$, and representing the new matrix as in (2.1), it will have the diagonal entries $\lambda_1, \lambda_3, \ldots, \lambda_{2^n-1}$ in order in the upper left and the diagonal entries $\lambda_2, \lambda_4, \ldots, \lambda_{2^n}$ in order in the lower right. To prove the proposition we use the following lemma.

LEMMA 2.2. *Let M and M' be density matrices and let $M \succeq M'$. Then $\iota\overline{w}M \succeq \iota M'$.*

Proof of Proposition 2.1. Consider any sequence of conjugations $\overline{u_1}, \ldots, \overline{u_r}$. Applying the lemma, induction on r shows that

$$\iota\, \overline{w}\, \iota\, \cdots\, \overline{w}\, \iota\, h \succeq \iota\, \overline{u_r}\, \iota\, \cdots\, \overline{u_1}\, \iota\, h. \tag{2.3}$$

Observe that for each r, the left-hand side of this expression is a diagonal density matrix. Hence each application of $\overline{w}$ is a classical operation, a permutation of the basis states. ∎

Proof of Lemma 2.2. For a density matrix

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{12}^\dagger & A_{22} \end{pmatrix}, \tag{2.4}$$

let $\alpha_1, \ldots, \alpha_{2^{n-1}}$ be the eigenvalues of $A_{11} + A_{22}$. Then the eigenvalues of ιA are $\frac{e^\varepsilon}{e^\varepsilon+e^{-\varepsilon}}\alpha_1, \frac{e^{-\varepsilon}}{e^\varepsilon+e^{-\varepsilon}}\alpha_1, \ldots, \frac{e^\varepsilon}{e^\varepsilon+e^{-\varepsilon}}\alpha_{2^{n-1}}, \frac{e^{-\varepsilon}}{e^\varepsilon+e^{-\varepsilon}}\alpha_{2^{n-1}}$. It follows that if another density matrix B is given (and partitioned in the same way) and if $A_{11} + A_{22} \succeq B_{11} + B_{22}$, then $\iota A \succeq \iota B$. So it remains to show that $(\overline{w}M)_{11} + (\overline{w}M)_{22} \succeq M'_{11} + M'_{22}$.

Let $\lambda_1 \geq \cdots \geq \lambda_{2^n}$ be the eigenvalues of M and let $\lambda'_1 \geq \ldots \geq \lambda'_{2^n}$ be the eigenvalues of M'. Then $(\overline{w}M)_{11}$ is the diagonal matrix with diagonal $(\lambda_1, \lambda_3, \ldots, \lambda_{2^n-1})$, and $(\overline{w}M)_{22}$ is the diagonal matrix with diagonal $(\lambda_2, \lambda_4, \ldots, \lambda_{2^n})$. The eigenvalues of $(\overline{w}M)_{11} + (\overline{w}M)_{22}$ are $(\lambda_1 + \lambda_2, \lambda_3 + \lambda_4, \ldots, \lambda_{2^n-1} + \lambda_{2^n})$; by the assumption that $M \succeq M'$, this majorizes the sequence $(\lambda'_1 + \lambda'_2, \lambda'_3 + \lambda'_4, \ldots, \lambda'_{2^n-1} + \lambda'_{2^n})$. It remains to show that the latter majorizes the eigenvalues of $M'_{11} + M'_{22}$.

A simple inequality (see [18, section 9G]) states that the eigenvalues of $M'_{11} + M'_{22}$ are majorized by the sequence $(\beta_1 + \gamma_1, \ldots, \beta_{2^{n-1}} + \gamma_{2^{n-1}})$, where $\beta_1 \geq \cdots \geq \beta_{2^{n-1}}$ are the eigenvalues of M'_{11} and $\gamma_1 \geq \cdots \geq \gamma_{2^{n-1}}$ are the eigenvalues of M'_{22}. The argument is completed by an inequality of Fan (see [18, section 9C]) which states that for any Hermitian H,

$$\begin{pmatrix} H_{11} & H_{12} \\ H_{12}^\dagger & H_{22} \end{pmatrix} \succeq \begin{pmatrix} H_{11} & 0 \\ 0 & H_{22} \end{pmatrix}; \tag{2.5}$$

applied to $H = M'$, this yields $(\lambda'_1, \lambda'_2, \ldots, \lambda'_{2^n}) \succeq (\beta_1, \ldots, \beta_{2^{n-1}}, \gamma_1, \ldots, \gamma_{2^{n-1}})$. ∎

We may therefore restrict our attention to classical cooling algorithms. Observe that every intermediate density matrix created by a classical algorithm is diagonal. Hence the classical cooling steps are equivalent to the following discrete process on probability distributions on the set $\{0,1\}^n$: begin with the uniform distribution on $\{0,1\}^n$. The only tool for modifying the probability distribution is "discrete cooling steps," which have the effect of transforming the current distribution (denoted p) to a new distribution (denoted p'), related to p by

$$\left.\begin{aligned} p'_{w0} &= (p_{w0} + p_{w1})\tfrac{e^\varepsilon}{e^\varepsilon+e^{-\varepsilon}} \\ p'_{w1} &= (p_{w0} + p_{w1})\tfrac{e^{-\varepsilon}}{e^\varepsilon+e^{-\varepsilon}} \end{aligned}\right\} \quad \begin{array}{l}\text{for each binary string} \\ w \text{ of length } n-1.\end{array} \tag{2.6}$$

There is no way of *directly* cooling the first $n-1$ bits, but in between cooling steps we can perform arbitrary permutations of the binary strings. In the discrete process, the role of a permutation of the basis states is to properly pair off the current probabilities before the next cooling step.

Due to Proposition 2.1, Theorem 1.1 is equivalent to showing that the above discrete process cannot increase any probability from its initial value, 2^{-n}, to any more than $2^{-n}e^{\varepsilon 2^{n-1}}$, while Theorem 1.4 is equivalent to showing that the discrete process cannot create a bit of constant bias in less than $\Omega(\tilde{\varepsilon}^{-2})$ cooling steps.

3. Preliminaries.

3.1. Special configurations. The set of probabilities of the basis states, $\{P(w) : w \in \{0,1\}^n\}$, will be referred to as the configuration of the computer.

Definition 3.1. *A "special" configuration is one of*

(a) *a configuration that can be created (out of any configuration, and by any pairing) by a cooling step;*

(b) *the starting configuration, in which all probabilities equal* 2^{-n}.

Note that in a special configuration of type (a), two states that were paired in the previous round, and now have probabilities p and p', satisfy $|\log p - \log p'| = 2\varepsilon$.

3.2. The PPA. If the basis states of the computer are relabeled so that their probabilities are $p_0 \geq \cdots \geq p_{2^n-1}$ (ties broken in arbitrary but fixed fashion), then for each even i we will refer to the states i and $i+1$ as each other's "partners."

The PPA or "partner-pairing algorithm" is simply the following process: In each cooling step, pair partners together. (This completely specifies the algorithm save only for the number of iterations.)

LEMMA 3.2. *In a special configuration, if states with probabilities p and p' are partners, then* $|\log p - \log p'| \leq 2\varepsilon$.

Proof. For the configuration of type (b) this is automatic; for those of type (a) let p be the probability of a state for which the lemma is violated and let q be the probability of the state with which it was paired in the previous round. Suppose $q > p$; the other case is similar. So p is now paired with a probability r for some $r < pe^{-2\varepsilon}$, and the interval (r, p) is empty of state probabilities. The interval $(-\infty, r]$ therefore contains only intact pairs from the previous round and hence an even number of state probabilities. So it cannot be that p's partner in this round is r. ∎

The next step in demonstrating Theorems 1.1 and 1.4 concerns the relation between the output of an arbitrary cooling algorithm B and that of the PPA.

COROLLARY 3.3. *Given any initial probability distribution* $p = \{p_0, \ldots, p_{2^n-1}\}$, *and any cooling algorithm B, the distribution which results from applying the PPA for r cooling steps majorizes the distribution which results from applying B for r cooling steps.*

Proof. This follows from Proposition 2.1 because the PPA is the restriction to probability distributions of the operator $\overline{w}$ defined in section 2. ∎

As a consequence, in pursuit of a bound on the maximum achievable probability of any one string, we can focus on the PPA. (The same lesson applies to any Schur-convex function of the probabilities, of which the maximum probability is but one example; see [18].)

4. Proof of Theorem 1.1.

4.1. Dynamics of cooling algorithms: Assemblies of chips. It is useful to apply the map $p \to \log(2^n p)$ to all the probabilities of a configuration, to obtain a set of 2^n "chips" arrayed on the real axis. Two chips at z_1 and z_2 which are paired by a cooling step are carried to two new chips at $T(z_1, z_2) \pm \varepsilon$, where T is given by

$$e^{T(z_1,z_2)+\varepsilon} + e^{T(z_1,z_2)-\varepsilon} = e^{z_1} + e^{z_2}. \tag{4.1}$$

We now need to understand more about the dynamics of the PPA. A central tool will be to designate certain subsets of the chips as *assemblies.* With an assembly S we associate a *center* $c(S)$ which is the arithmetic mean of the chips, a *radius* $r(S)$ which is $\varepsilon/2$ times the number of chips in the assembly, and an *interval* I_S which is the closed interval $[c(S) - r(S), c(S) + r(S)]$. (We define assemblies only for special configurations.) A set of chips qualifies as an assembly if either

1. it is a pair of chips z_1 and z_2 which are partners (note that the center of this assembly is $(z_1 + z_2)/2$ and its radius is ε);
2. it is the union of two assemblies whose intervals intersect (we will refer to this as merging the two assemblies).

A maximal assembly is one which cannot be merged with any other assembly.

4.2. The modified PPA. The nonlinear map T (defined in (4.1)) is difficult to work with directly, but it has a linearization which suits our needs. In the modified process, chips at z_1 and z_2 are carried to the pair $M(z_1, z_2) \pm \varepsilon$, where

$$M(z_1, z_2) = (z_1 + z_2)/2. \tag{4.2}$$

(The modified process does not preserve the identity $2^{-n} \sum e^{z_i} = 1$.) In the modified PPA, partners are defined among the chips just as before, but the map M rather than the map T is applied to each pair. That the modified process is a useful approximation to the true process is due to the twin facts that ε is small and that in a special configuration, partners z_1 and z_2 are close. The bearing of the modified process on the true process is expressed in the following lemma.

LEMMA 4.1. *Consider two sets of chips in special configurations $x_0 \geq \cdots \geq x_{2^n-1}$ and $y_0 \geq \cdots \geq y_{2^n-1}$, such that $x_i \leq y_i$ for all i. Apply a step of the true PPA to $x_0, \ldots, x_{2^n-1}$, resulting in the set of chips $x'_0 \geq \cdots \geq x'_{2^n-1}$. Apply a step of the modified PPA to $y_0, \ldots, y_{2^n-1}$, resulting in the set of chips $y'_0 \geq \cdots \geq y'_{2^n-1}$. Then $x'_i \leq y'_i$ for all i.*

Proof. We have only to show that for any even i, $T(x_i, x_{i+1}) \leq M(y_i, y_{i+1})$. We have that

$$e^{T(x_i,x_{i+1})} = e^{M(x_i,x_{i+1})} \frac{\cosh((x_i - x_{i+1})/2)}{\cosh(\varepsilon)}. \tag{4.3}$$

Since the configuration $x_1, \ldots, x_{2^n}$ is special, $|x_i - x_{i+1}| \leq 2\varepsilon$ by Lemma 3.2, and so $T(x_i, x_{i+1}) \leq M(x_i, x_{i+1})$. Since $M(x_i, x_{i+1}) \leq M(y_i, y_{i+1})$ directly from the assumptions, we conclude that $T(x_i, x_{i+1}) \leq M(y_i, y_{i+1})$. ∎

Applying each of the processes T and M repeatedly starting from a common special configuration, we conclude by induction that after any number of iterations, the greatest achievable probability of any state in the modified process is an upper bound on the probability of any state in the true process.

The chip game. The modified process given by (4.2) describes the following chip game: 2^n chips are placed initially at the origin of the real line. In each step you choose a pairing of the chips, and then the positions of each pair of chips (say z_1 and z_2) are moved to $(z_1 + z_2)/2 \pm \varepsilon$. Your goal is to move any one chip as far to the right as possible. Theorem 1.1 has been reduced to showing that no chip can be moved to distance more than $\varepsilon 2^{n-1}$ from the origin. Section 4.3 is devoted to a somewhat lengthy combinatorial proof of this fact.

Fortunately, there is a simpler proof of a bound that is weaker by a factor of 2: i.e., no chip can be moved to distance more than $\varepsilon 2^n$ from the origin. This is sufficient to establish our fundamental physical conclusions—to wit: unbounded cooling is impossible using finitely many computation qubits at a fixed heat-bath temperature; moreover, for large n there is a threshold at $(-\log_2 \tilde{\varepsilon})/n = 1$ for the feasibility of cooling. The reader interested only in these conclusions can read the following and skip section 4.3.

The bound of $\varepsilon 2^n$ rests on showing that the modified PPA, starting from the initial configuration having all chips at the origin, never creates a separation of more than 2ε between adjacent chips: Suppose the gaps within a set of chips $x_0 \geq \cdots \geq x_{2^n-1}$ are bounded by 2ε, and that a step of the modified PPA is applied to these chips, carrying x_{2i} to $x'_{2i} = (x_{2i} + x_{2i+1})/2 + \varepsilon$, and x_{2i+1} to $x'_{2i+1} = (x_{2i} + x_{2i+1})/2 - \varepsilon$. Note that for even i, $x'_i \geq x_i$, while for odd i, $x'_i \leq x_i$. The $\{x'_i\}$ are generally not in sorted order, but x'_{2^n-1} is a smallest chip, and so it is enough to show that for every

$i < 2^n - 1$, $x'_{i+1} \geq x'_i - 2\varepsilon$ (i.e., there are no descents by more than 2ε in the sequence $x'_0, \ldots, x'_{2^n-1}$). For even i, $x'_{i+1} \geq x'_i - 2\varepsilon$ is satisfied with equality. For odd i, using the inductive hypothesis, $x'_{i+1} \geq x_{i+1} \geq x_i - 2\varepsilon \geq x'_i - 2\varepsilon$.

Finally, a configuration of chips whose mean is 0 and in which all gaps are bounded by 2ε has no chip beyond distance $\varepsilon 2^n$ from the origin. For if any gap is less than 2ε, the configuration does not achieve greatest possible distance, since the chips to the right and left of this gap can be shifted outward while preserving the mean, while a configuration in which all the gaps are exactly 2ε is an arithmetic sequence centered at the origin.

We return to the proof of the full statement of Theorem 1.1.

4.3. Preservation of maximal assemblies. The most lengthy technical portion of this paper goes into establishing the following proposition.

PROPOSITION 4.2.

1. *The maximal assemblies of a special configuration partition the set of chips. (Equivalently, we can arrive at the list of all maximal assemblies by merging assemblies in any order until no further mergers are possible.)*
2. *Maximal assemblies are preserved by the modified PPA (i.e., the partition of the chips of a special configuration into maximal assemblies is unchanged by a cooling step).*

We begin with a sequence of arguments that do not depend on whether the true or modified PPA is applied but only on the fact that each step pairs partners together.

LEMMA 4.3. *In a special configuration, if $a, b \in \mathbb{R}$, $a \leq b$, and the intervals $[a - 2\varepsilon, a)$ and $(b, b + 2\varepsilon]$ are empty of chips, then the interval $[a, b]$ contains an even number of chips.*

Proof. In a special configuration, two chips that were paired in the previous round are separated by 2ε. The fact that $[a - 2\varepsilon, a)$ is empty of chips therefore implies that there are an even number of chips in $(-\infty, a)$; similar reasoning shows there are an even number of chips in (b, ∞). □

LEMMA 4.4. *Let $k \geq 0$, k even, and let D be an assembly of cardinality at most k.*

1. *(Bounded gap). If $S \subseteq D$ consists of some of the partner pairs of D, then $I_S \cap I_{D-S} \neq \emptyset$.*
2. *(Monotonicity). If an assembly B is a subset of D, then I_B is contained in I_D.*

Proof. The proof is by induction on k.

Part 1, Bounded gap: Let $P_1, \ldots, P_{k/2}$ be the partner pairs of D, listed in their order on the line. Suppose the lemma fails for $S = P_1 \cup \cdots \cup P_\ell$, for some $0 < \ell < k/2$. Fix a sequence of mergers that forms D out of $P_1, \ldots, P_{k/2}$. We may assume these mergers always combine adjacent assemblies, since if an assembly B is between A and C which are being merged, I_B must intersect one of I_A or I_C (say I_A); B can be merged with A. By part 2 of the lemma (for $k - 2$), all subsequent merger steps which are supposed to be performed with the assembly containing A or B can still be performed (in particular the very next step of merging $A \cup B$ with C). So the mergers describe a binary tree T_0 of assemblies, whose leaves are the partner pairs and whose internal nodes are the assemblies constructed during the merging process; the left and right children of any internal node are always two disjoint assemblies which are adjacent to each other in the left-right order on the line. Moreover, the children of any internal node are two disjoint assemblies whose intervals intersect, since this is a tree of mergers. The root of T_0 is D.

We will also use other trees in the proof. The internal nodes of these trees might not be assemblies, but each internal node will still be a sequence of partner pairs that are laid out consecutively on the line; the left and right children of an internal node will still consist of two sequences which are disjoint, adjacent to each other on the line, and in the same left-right order. While the set at an internal node may not be an assembly, we will still associate with such a set of partner pairs a center, a radius, and an interval, all defined just as they are for an assembly.

We will say that an internal node is "cohesive" if the intervals of its two children intersect. Every internal node in T_0 is cohesive. The claim will follow from the existence of another tree T_F in which the left child of the root is S, the right child is $D - S$, and the root is cohesive.

We will show the existence of T_F by converting T_0 into it through a sequence of "tree rotations." In a tree rotation a tree T' is changed into a tree T'' as follows. Let A, B, and C each be a sequence of consecutive partner pairs, and let these sequences be disjoint, and arranged adjacent to each other on the line from left to right in the order A, B, C. Suppose that each occurs as a node in T' and that there are internal nodes $A \cup B$ and $A \cup B \cup C$. Then a right tree rotation "at $A \cup B \cup C$" is the conversion of T' into the tree T'' that differs only in that instead of an internal node $A \cup B$, it has an internal node $B \cup C$. (A left tree rotation would be the replacement of a node $B \cup C$ by a node $A \cup B$.) We will demonstrate the following property of right tree rotations; the analogous property holds for left tree rotations and is shown in the same way.

(*) If $A \cup B$ and $A \cup B \cup C$ are cohesive in T', then $A \cup B \cup C$ is cohesive in T''.

Using (*) we will obtain the desired tree T_F by beginning with T_0, in which all internal nodes are cohesive, and repeatedly doing the following: Find the least common ancestor J of P_ℓ and $P_{\ell+1}$, let K be its parent, and rotate at K. After the rotation, K becomes the new least common ancestor of P_ℓ and $P_{\ell+1}$; by (*), it is still cohesive. The cohesiveness of nodes outside the subtree rooted at K is unaffected by the rotation. Hence the process continues until a last rotation at the root, at which time the root is cohesive, and is the least common ancestor of P_ℓ and $P_{\ell+1}$.

Finally, we show (*). For simplicity of notation and without loss of generality we will assume the center of B is 0. Let A have center $-r_1$ and radius s_1; let B have radius s_2; and let C have center r_3 and radius s_3. Note $s_1, s_2, s_3, r_1, r_3 \geq 0$. Cohesiveness of $A \cup B$ in T' means that

$$r_1 \leq s_1 + s_2, \tag{4.4}$$

while cohesiveness of $A \cup B \cup C$ in T' means that

$$r_3 + \frac{r_1 s_1}{s_1 + s_2} \leq s_1 + s_2 + s_3. \tag{4.5}$$

Sum these inequalities with the respective nonnegative coefficients $\frac{s_2(s_1+s_2+s_3)}{(s_1+s_2)(s_2+s_3)}$ and $\frac{s_3}{(s_1+s_2)(s_2+s_3)}$ to obtain

$$r_1 + \frac{r_3 s_3}{s_2 + s_3} \leq s_1 + s_2 + s_3, \tag{4.6}$$

which indicates the cohesiveness of $A \cup B \cup C$ in T''.

Part 2, Monotonicity: The proof is in two sections. (a) We argue that we can form D in a sequence of strict mergers that create B as an intermediate step. (A

strict merger is one that forms the union of two assemblies neither of which contains the other.) (b) We argue that in any strict merger, forming assembly $C = B_1 \cup B_2$ from B_1 and B_2, the interval of C contains those of B_1 and B_2.

Proof of (a). First, carry out the mergers that create B from the original pairs. Now consider the mergers that create D from the original pairs. Carry out those steps, each time replacing the arguments E_1 and E_2 of a desired merger $E_1 \cup E_2$, with the present (greatest) assemblies that contain E_1 and E_2. We must check that this makes sense: that each of E_1 and E_2 are a subset of a present assembly, and that the intervals of those assemblies intersect. The latter claim holds by induction because, until D has been formed, those assemblies are smaller than D (and because after D has been formed, all mergers are trivial). To see the former claim, observe (by induction on the step number) that at any time, the present greatest assembly containing E_i is either E_i, or $E_i \cup B$, depending on whether any of the pairs in E_i intersects B.

Proof of (b). Let $s = |B_1 \cap B_2|$, $s_1 = |B_1 - B_2|$, and $s_2 = |B_2 - B_1|$. Let c be the arithmetic mean of $B_1 \cap B_2$, c_1 the arithmetic mean of $B_1 - B_2$, and c_2 the arithmetic mean of $B_2 - B_1$.

The arithmetic mean of B_1 is $\overline{c}_1 = (cs + c_1 s_1)/(s + s_1)$, and the arithmetic mean of B_2 is $\overline{c}_2 = (cs + c_2 s_2)/(s + s_2)$. Let $\overline{c}$ be the arithmetic mean of $B_2 \cup B_1$, so $\overline{c} = (cs + c_1 s_1 + c_2 s_2)/(s + s_1 + s_2)$. To demonstrate the containment of intervals, we show that the left-hand boundary of I_C, $\overline{c} - s - s_1 - s_2$, is to the left of the left-hand boundary of I_{B_1}, $\overline{c}_1 - s - s_1$; in other words, $\overline{c}_1 - s - s_1 \geq \overline{c} - s - s_1 - s_2$. The remaining three cases are similar.

By part 1 of the lemma we know

$$s + s_2 \geq c_2 - c, \tag{4.7}$$

$$s + s_1 \geq c - c_1. \tag{4.8}$$

Inequality (4.7) is equivalent to $c_2 - \overline{c}_1 \leq s + s_2 + c - \overline{c}_1$. Inequality (4.8) is equivalent to $c - \overline{c}_1 \leq s_1$. Together these give

$$c_2 - \overline{c}_1 \leq s + s_1 + s_2, \tag{4.9}$$

which is equivalent to

$$\overline{c} - \overline{c}_1 \leq s_2. \tag{4.10}$$ □

We now prove Proposition 4.2(1): *The maximal assemblies of a special configuration partition the set of chips. (Equivalently, we can arrive at the list of all maximal assemblies by merging assemblies in any order until no further mergers are possible.)*

Proof. The initial pairing of chips is fixed. Let $S_1, \ldots, S_k$ be the maximal assemblies obtained by a particular sequence of mergers. Write, in terms of the initial pairs, $S_1 = P_{11} \cup \cdots \cup P_{1\ell_1}$, $S_2 = P_{21} \cup \cdots \cup P_{2\ell_1}$, and so forth. Fixing an alternate merger sequence, consider the first step in which that sequence joins pairs from some two different S_i's; suppose those are S_1 and S_2. Let $S_1' \subseteq S_1$ and $S_2' \subseteq S_2$ be the two assemblies merged in this step. Then the intervals of S_1' and S_2' intersect, which contradicts Lemma 4.4(2), since the intervals of S_1 and S_2 do not intersect. □

COROLLARY 4.5. *Let D be an assembly and let $S \subseteq D$ be a set of even cardinality. Then I_S is contained in I_D.*

Proof. We show that the right end of I_S is less than the right end of I_D; a similar argument shows that the left end of I_S is greater than the left end of I_D. The set S' consisting of the rightmost $|S|$ chips in D consists of several partner pairs. By Lemma 4.4(1), $I_{S'}$ intersects $I_{D-S'}$; this implies that $I_{S'} \subseteq I_D$. ∎

LEMMA 4.6. *Given a cooling step, form a corresponding set of intervals S as follows. For each two chips paired by the cooling step, S contains the interval between the poststep positions of those chips. Also, for each pair of partners in the poststep configuration, S contains the interval between the partners. Consider any point that coincides with no poststep chips. Then there is an even number of intervals of S containing that point.*

Proof. Moving from left to right, at every chip the number of partner intervals covering the line alternates between 0 and 1. The parity of the contribution of the pair intervals also alternates at every chip. Therefore, between chips, the parity is the same as it is beyond the last chip: 0. ∎

LEMMA 4.7. *Two chips which are paired in a cooling step (of any algorithm) are in a common maximal assembly after that cooling step.*

Proof. For specificity suppose this step was numbered t. Let x be such that the positions of the two chips after step t are $x \pm \varepsilon$. We will use the terms "righties" and "lefties" to refer to members of pairs depending on whether they are, respectively, the higher or lower probability chip (after the step); e.g., $x + \varepsilon$ is the righty (or t-righty, to specify the step) and $x - \varepsilon$ is the t-lefty of their pair.

We consider several cases.

Case 1. $x \pm \varepsilon$ are partners poststep. The lemma follows.

Otherwise, for $-\varepsilon \le s_1, s_2 \le \varepsilon$, let $x - \varepsilon$ be partnered with a chip at $x - \varepsilon + 2s_1$ and let $x+\varepsilon$ be partnered with a chip at $x+\varepsilon+2s_2$. The number of chips, m, between the pairs $\{x-\varepsilon, x-\varepsilon+2s_1\}$ and $\{x+\varepsilon, x+\varepsilon+2s_2\}$ is even; we consider several cases.

Case 2. $s_1 \ge s_2$. In this case the assembly $\{x-\varepsilon, x-\varepsilon+2s_1\}$ (whose right-hand boundary is at $x+s_1$) and the assembly $\{x+\varepsilon, x+\varepsilon+2s_2\}$ (whose left-hand boundary is at $x + s_2$) intersect geometrically, and the lemma follows.

Case 3. $s_1 < s_2$. We start by showing that $m > 0$, which is to say that the interval $[x-\varepsilon+2s_1, x+\varepsilon+2s_2]$ contains chips other than $x-\varepsilon$ or $x+\varepsilon$. The interval between $\max\{x-\varepsilon, x-\varepsilon+2s_1\}$ and $\min\{x+\varepsilon, x+\varepsilon+2s_2\}$ is nonempty, and since it is contained in $[x-\varepsilon, x+\varepsilon]$, it must by Lemma 4.6 intersect some interval between two chips that were paired in the last cooling step; neither $x-\varepsilon+2s_1$ nor $x+\varepsilon+2s_2$ can be one of the chips generating such an interval, since the distance between them is greater than 2ε. Hence m is positive; we continue with two cases depending on its value.

Case 3a. $s_1 < s_2$ and $m = 2$. Let the two points be z_1 and z_2, with $z_1 \le z_2$; note that these are paired together poststep and that $x-\varepsilon \le z_1 \le z_2 \le x+\varepsilon$. By the parity argument of Lemma 4.6, there must be two chips that were paired in step t, for which the interval between the poststep chip positions covers the interval between $x-\varepsilon+2s_1$ and z_1; therefore $z_1 \le x+\varepsilon+2s_1$. For a similar reason, $z_2 \ge x-\varepsilon+2s_2$. We examine three pair assemblies: $A = \{x-\varepsilon, x-\varepsilon+2s_1\}$, $B = \{z_1, z_2\}$, and $C = \{x+\varepsilon, x+\varepsilon+2s_2\}$. Their intervals are $I_A = [x-2\varepsilon+s_1, x+s_1]$, $I_B = [(z_1+z_2)/2-\varepsilon, (z_1+z_2)/2+\varepsilon]$, and $I_C = [x+s_2, x+2\varepsilon+s_2]$. If I_B does not intersect I_A, then $x+s_1+\varepsilon < (z_1+z_2)/2$. If in addition I_B does not intersect I_C, then $(z_1+z_2)/2 < x+s_2-\varepsilon$, together implying $s_1+2\varepsilon < s_2$, which is impossible, since $-\varepsilon \le s_1, s_2 \le \varepsilon$. Hence I_B intersects at least one of I_A or I_C. The rest of the argument is symmetric for these two cases, and so we spell out only the case that I_B intersects I_A.

If I_B intersects I_A, the four points of A and B form an assembly $A \cup B$ whose right-hand boundary is at $(2(x-\varepsilon)+2s_1+z_1+z_2)/4+2\varepsilon$ which, by the lower bounds for z_1 and z_2, is at least $x+\varepsilon+(s_1+s_2)/2$. Subtracting the left-hand boundary of I_C gives $\varepsilon+(s_1-s_2)/2$, which by the constraints on s_1 and s_2 is at least 0. Hence the interval of the assembly $A \cup B$ intersects that of the assembly C, and the lemma follows.

Case 3b. $s_1 < s_2$ and $m \geq 4$. In this case there are at least four points $z_1, \ldots, z_m$ arranged as $x-\varepsilon \leq z_1 \leq \cdots \leq z_m \leq x+\varepsilon$; the same argument used for Case 3a shows that either the assembly of $\{z_1, z_2\}$ intersects that of $\{x-\varepsilon, x-\varepsilon+2s_1\}$, or the assembly of $\{z_{m-1}, z_m\}$ intersects that of $\{x+\varepsilon, x+\varepsilon+2s_2\}$. The cases are symmetric, and so suppose that the first of these occurs. Then the assembly formed by $D = \{z_1, z_2, x-\varepsilon, x-\varepsilon+2s_1\}$ has its right-hand boundary at $(2(x-\varepsilon)+2s_1+z_1+z_2)/4+2\varepsilon$. Using the lower bound $x-\varepsilon$ for z_1 and z_2 places a lower bound of $x+\varepsilon+s_1/2$ on this boundary. For the interval of the assembly $\{z_{m-1}, z_m\}$ not to intersect this, we must have $(z_{m-1}+z_m)/2 > x+s_1/2+2\varepsilon$. The right-hand boundary of the assembly $\{z_{m-1}, z_m\}$ must therefore be at a position greater than $x+s_1/2+3\varepsilon$, which in turn is at least $x+5\varepsilon/2$. Hence the four points $E = \{z_{m-1}, z_m, x+\varepsilon, x+\varepsilon+2s_2\}$ form an assembly. Using the upper bound $x+\varepsilon$ on z_{m-1} and z_m places an upper bound of $x-\varepsilon+s_2/2$ on the left-hand boundary of this assembly. The intervals of the assemblies D and E intersect because $(x+\varepsilon+s_1/2)-(x-\varepsilon+s_2/2) = 2\varepsilon+(s_1-s_2)/2 \geq \varepsilon \geq 0$. The lemma follows. ∎

We can now finally prove Proposition 4.2(2): *Maximal assemblies are preserved by the modified PPA.*

We show, equivalently, that every prestep assembly is contained in a poststep assembly. Lemma 4.7 establishes this for prestep pairs. Now suppose that the prestep assembly D was formed by merging assemblies B and C. By induction B and C are each contained in a poststep assembly; call these B' and C'. By Corollary 4.5, the intervals of B' and C' contain those of the poststep sets of chips B and C. In the modified chip process, these last two intervals are identical, respectively, to the intervals of the prestep assemblies B and C. Therefore $I_B \subseteq I_{B'}$ and $I_C \subseteq I_{C'}$. Since I_B intersects I_C, $B' \cup C'$ is an assembly, and it contains D. ∎

Proof of Theorem 1.1. Proposition 4.2 implies that in a configuration reachable from the start state by the modified chip process there is just a single maximal assembly, whose interval is $[-\varepsilon 2^{n-1}, \varepsilon 2^{n-1}]$. Consider a chip that is furthest from the origin: by Lemma 3.2, it lies within the interval of the assembly formed by itself and its partner; by Lemma 4.4(2), this interval is contained within the interval of the maximum assembly. Hence all chips lie within distance $\varepsilon 2^{n-1}$ of the origin. Due to Corollary 3.3 (applied to the initial uniform distribution which corresponds to the maximally mixed state $\mathcal{I}_n$) and Lemma 4.1, this shows that no cooling process can increase any probability above $2^{-n}e^{\varepsilon 2^{n-1}}$, establishing the theorem. ∎

5. Proof of Theorem 1.2. Here we prove Theorem 1.2, which is a complement to the "impossibility" result of Theorem 1.1: *For $\tilde{\varepsilon} \geq 2^{4-n}$, heat-bath cooling using the PPA produces a distribution at variation distance $\Theta(1)$ from uniform, within $T = \tilde{\varepsilon}^{-2}$ cooling steps.* To a state with probability p assign the potential $g(p) = \log\cosh(2^n(p-2^{-n}))$, and to a configuration c assign the potential $g(c) = \sum_p g(p)$. Observe that $g(\text{initial configuration}) = 0$.

Let c be any special configuration and let c' be the configuration it is carried to by the PPA. Let $\Delta g(c) = g(c') - g(c)$. If p_1 and p_2 are paired in c, then their

contribution to $\Delta g(c)$ is

$$g(p_1') + g(p_2') - g(p_1) - g(p_2), \tag{5.1}$$

where without loss of generality $p_1 \le p_2$, and we have written $p_1' = (p_1+p_2)(1-\tilde{\varepsilon})/2$ and $p_2' = (p_1+p_2)(1+\tilde{\varepsilon})/2$. This is nonnegative because g is convex, $p_1+p_2 = p_1'+p_2'$, and because due to Lemma 3.2, $[p_1,p_2] \subseteq [p_1',p_2']$.

Since g is strictly convex, the potential of a special configuration increases strictly unless each of its pairs $\{p_1,p_2\}$ satisfies $|\log p_2 - \log p_1| = 2\varepsilon$.

If sometime within T rounds it happens that there are at least $2^{n-1}-2$ probabilities outside of the interval $[2^{-n-1}, 3\cdot 2^{-n-1}]$, then we are done.

Otherwise, suppose that c is a special configuration having at least $2^{n-1}+2$ probabilities within the interval $[2^{-n-1}, 3\cdot 2^{-n-1}]$. We want to show a lower bound on $\Delta g(c)$. The PPA must form at least 2^{n-2} pairs among these probabilities , and at least 2^{n-3} of those pairs must be of length (separation between the probabilities) at most 2^{3-2n}. The contribution of such a pair to $\Delta g(c)$ is least if its length is indeed 2^{3-2n}; for a lower bound on $\Delta g(c)$ we also assume that the poststep probabilities are as close to each other as possible, which (since their ratio is fixed at $e^{2\varepsilon}$) occurs when the average of the probabilities is as small as possible, namely $2^{-n-1}\cdot 2\tilde{\varepsilon} = 2^{-n}\tilde{\varepsilon}$. Letting the probabilities of the pair, before the cooling step, be $y \pm 2^{2-2n}$, and letting Δ_1 be the contribution of these two probabilities to $\Delta g(c)$, we can write

$$\Delta_1 = \log \frac{\cosh 2^n(y(1+\tilde{\varepsilon}) - 2^{-n}) \cosh 2^n(y(1-\tilde{\varepsilon}) - 2^{-n})}{\cosh 2^n(y + 2^{2-2n} - 2^{-n}) \cosh 2^n(y - 2^{2-2n} - 2^{-n})}. \tag{5.2}$$

Let $x = 2^n y - 1$; note that $|x| \le 1/2$. Let $\eta = 2^n y\tilde{\varepsilon}$; since $y \ge 2^{-n-1}$, $\eta \ge \tilde{\varepsilon}/2$.

$$\Delta_1 = \log \frac{\cosh(x+\eta)\cosh(x-\eta)}{\cosh(x+2^{2-n})\cosh(x-2^{2-n})} = \log \frac{\cosh 2x + \cosh 2\eta}{\cosh 2x + \cosh 2^{3-n}}. \tag{5.3}$$

Since this is increasing in η for $\eta > 0$, we have

$$\Delta_1 \ge \log \frac{\cosh 2x + \cosh \tilde{\varepsilon}}{\cosh 2x + \cosh 2^{3-n}}. \tag{5.4}$$

Now let $h(z) = \log \frac{\cosh 2x + \cosh(\tilde{\varepsilon}/2+z)}{\cosh 2x + \cosh 2^{3-n}}$. Then $\Delta_1 \ge h(\tilde{\varepsilon}/2)$. Now

$$h'(z) = \frac{\sinh(\tilde{\varepsilon}/2+z)}{\cosh 2x + \cosh(\tilde{\varepsilon}/2+z)}, \tag{5.5}$$

$$h''(z) = \frac{1+\cosh(\tilde{\varepsilon}/2+z)\cosh 2x}{(\cosh 2x + \cosh(\tilde{\varepsilon}/2+z))^2} \ge 0. \tag{5.6}$$

Thus $h(z) \ge h(0) + zh'(0)$, and in particular, since $\tilde{\varepsilon}/2 \ge 2^{3-n}$, $h(0) \ge 0$, and

$$\Delta_1 \ge h\left(\frac{\tilde{\varepsilon}}{2}\right) \ge 0 + \frac{\tilde{\varepsilon}}{2}\frac{\sinh(\tilde{\varepsilon}/2)}{\cosh 2x + \cosh(\tilde{\varepsilon}/2)} \ge \frac{\tilde{\varepsilon}^2}{8\cosh 1}, \tag{5.7}$$

the last inequality being implied by $|x| \le 1/2$, $\tilde{\varepsilon} \le 1$, and $\sinh(\tilde{\varepsilon}/2) \ge \tilde{\varepsilon}/2$. Consequently, if for T rounds it does not occur that at least $2^{n-1}-2$ probabilities are

outside of the interval $[2^{-n-1}, 3 \cdot 2^{-n-1}]$, then due to the 2^{n-3} pairs to which this analysis applies, g increases to at least

$$\frac{T 2^{n-6} \tilde{\varepsilon}^2}{\cosh 1}. \tag{5.8}$$

Observe now that for any p, $|p - 2^{-n}| \geq g(p) 2^{-n}$. So the variation distance from uniform after $T = \tilde{\varepsilon}^{-2}$ steps rises to at least

$$\frac{2^{-6}}{\cosh 1}. \tag{5.9}$$ □

6. Proof of Theorem 1.3. Here we prove Theorem 1.3, the second form (and the one more directly relevant to quantum computation) of the complement to the "impossibility" result of Theorem 1.1. *Within* $4n\tilde{\varepsilon}^{-2}(1 + \log(1/\tilde{\varepsilon}))$ *cooling steps, the PPA creates a probability distribution in which with probability at least* $1 - O(\frac{1}{1+\log 1/\tilde{\varepsilon}})$, *all of the first* $n - (1 + o(1)) \log_2 1/\tilde{\varepsilon}$ *bits are* $|0\rangle$*'s.*

Proof. As in the previous section we use a potential function, but now we use a different function—the entropy of the distribution—and we use it only for the runtime analysis, rather than using low entropy to imply that many cold bits are extracted.

Let $\mathcal{H}$ be the entropy function, and for $0 \leq \delta \leq 1$ let $H(\delta) = \mathcal{H}(\{(1-\delta)/2, (1+\delta)/2\}) = \frac{1-\delta}{2} \log \frac{2}{1-\delta} + \frac{1+\delta}{2} \log \frac{2}{1+\delta}$. Let $(1 \pm \delta)p/2$ be two probabilities paired in a cooling step. The change in their contribution to the distribution entropy due to the cooling step is $(H(\tilde{\varepsilon}) - H(\delta))p$; due to Lemma 3.2, $\delta \leq \tilde{\varepsilon}$, and so this contribution is nonpositive. Thus the distribution entropy is weakly decreasing in each cooling step.

LEMMA 6.1. *Within* $\frac{n \log 2}{(H(\delta) - H(\tilde{\varepsilon}))\gamma}$ *cooling steps, at least* $1 - \gamma$ *of the probability resides in partners* $\{p_1, p_2\}$ *for which* $|\log p_1 - \log p_2| \geq 2\delta$.

Proof. So long as the condition is unfulfilled, at least γ of the probability resides in partners for which $|\log p_1 - \log p_2| \leq 2\delta$, and so the distribution entropy (which begins as $n \log 2$) decreases in each cooling step by at least $(H(\delta) - H(\tilde{\varepsilon}))\gamma$. □

LEMMA 6.2. *If at least* $1 - \gamma$ *of the probability resides in partners* $\{p_1, p_2\}$ *for which* $|\log p_1 - \log p_2| \geq 2\delta$, *then for positive even* y, *at least* $(1-\gamma)(1 - e^{-(y+2)\delta})$ *of the probability resides in just* y *of the states.*

Proof. The probability of the y most likely states is at least equal to the probability of the y most likely states in partner pairs for which $|\log p_1 - \log p_2| \geq 2\delta$. That probability is maximized by the distribution in which the partners pairs are adjacent, which is to say that each probability occurs twice (except at the ends), once as the smaller and once as the larger of two partners. A short calculation shows that the sum of the top y probabilities is at least $(1-\gamma)(1 - e^{-(y+2)\delta})$. □

Finally, we can establish Theorem 1.3. Let $\gamma = \frac{\log 2}{1+\log 1/\tilde{\varepsilon}}$, $y = \frac{2 \log 1/\gamma}{\tilde{\varepsilon}}$, and $\delta = \tilde{\varepsilon}/2$. The total probability of these y most likely states is $1 - O(\frac{1}{1+\log 1/\tilde{\varepsilon}})$, and once indexed lexicographically in decreasing likelihood from 0 to $2^n - 1$, they all share $|0\rangle$'s in their first $n - \lg y \geq n - (1 + o(1)) \lg 1/\tilde{\varepsilon}$ bits. □

7. Proof of Theorem 1.4. We demonstrate here the lower bound of $\Omega(\tilde{\varepsilon}^{-2})$ on the number of cooling steps required in order to create even a single bit of constant bias. As in section 6, we examine the entropy of the distribution. The initial entropy is $n \log 2$. A distribution in which some bit has bias bounded away from 0 has entropy $(n - \Omega(1)) \log 2$. From the calculations in section 6 we see that the entropy of the distribution can decrease by at most $\log 2 - H(\tilde{\varepsilon}) \leq \tilde{\varepsilon}^2$ in a single cooling step. Hence a total of $\Omega(\tilde{\varepsilon}^{-2})$ cooling steps is required. □

8. Proof of Theorem 1.5. To this point we have concentrated on what can be achieved by alternating *arbitrary* permutations with cooling steps. It is not known whether the quality of initialization achieved in Theorems 1.2 and 1.3 can also be efficiently produced if the permutations must be implemented with one of the standard bases of reversible gates. However, we now outline why slightly weaker cooling can indeed be achieved using a simple sequence of standard reversible gates. Theorems 1.2 and 1.3 guarantee good initialization, provided, respectively, that $\tilde{\varepsilon} \geq 2^{4-n}$ and $\tilde{\varepsilon} \in \Omega(n2^{-n})$; the simple procedure provided in this section initializes a bit with bias $\Omega(1)$ within time $O((1/\tilde{\varepsilon})^{\log\log 1/\tilde{\varepsilon}})$, provided that $\tilde{\varepsilon} \in \Omega(\phi^{-n})$. (Recall that $\phi = (1+\sqrt{5})/2$.) More generally, for $N \leq \min\{n, \lceil \log_\phi 1/\tilde{\varepsilon} \rceil\}$, the procedure prepares a bit of bias $\Omega(\tilde{\varepsilon}F_N)$ within time $\exp(O(N \log N))$ using an N-bit device. In what follows set $N = \min\{n, \lceil \log_\phi 1/\tilde{\varepsilon} \rceil\}$.

Recall that $F_k = (2\phi/5 - 1/5)\phi^k - (2\phi/5 - 1/5)(1-\phi)^k$. For notational convenience we assume in this section that bit 1 (rather than n) is the special bit that can be directly cooled by the heat bath.

The procedure $\mathcal{F}$, taking argument $2 \leq N \leq \lceil \log_\phi 1/\tilde{\varepsilon} \rceil$, produces statistically independent bits $1, \ldots, N$, such that bit k (for every $1 \leq k \leq N$) has bias $\approx \tilde{\varepsilon}F_k$, or more specifically, bias $\geq \tilde{\varepsilon}F_k(1 - 2^{k-N-1})$. The sequence of quantum gates to be applied in this simple recursive procedure is easily generated by an NC1 circuit (whose input is the elapsed time in the cooling procedure).

Procedure $\mathcal{F}(N)$*:* Run $\mathcal{F}'(N, N)$.

Procedure $\mathcal{F}'(N, k)$*:*

(a) If $k = 2$, run the cooling step on bits 1 and (by exchange) 2.

(b) If $k > 2$, repeat steps (b1) and (b2) $O(N - k)$ times until the bias of bit k is at least $\tilde{\varepsilon}F_k(1 - 2^{k-N-1})$:

  (b1) Use a reversible majority gate to set bit k to be the majority of bits $k-2, k-1$, and k.

  (b2) Run $\mathcal{F}'(N, k-1)$.

(There are various ways to implement a reversible majority gate. Conceptually perhaps the simplest is the transformation of a triple of bits (a, b, c) into the triple $(\mathrm{MAJ}(a, b, c), a \oplus b, a \oplus c)$.)

We start with an imprecise version of the analysis. The effect of the majority gate in (b1) is, roughly, to transform bits with biases $\tilde{\varepsilon}\phi^{k-2}, \tilde{\varepsilon}\phi^{k-1}$, and $\tilde{\varepsilon}x$ into a bit of bias $\approx \tilde{\varepsilon}(\phi^{k-2} + \phi^{k-1} + x)/2$ (this is an approximation accurate for biases $\ll 1$). In each iteration within step (b), x converges toward the unique fixed point of this transformation, $x = \phi^k$. Convergence of the loop inside step (b) is rapid: in each iteration, the Lyapunov function $(\phi^k - x)^2$ decreases by a factor of almost 4. (When step (b) is very close to completion the factor is no longer close to 4 but remains bounded away from 1.) This is why $O(N - k)$ repetitions are enough.

The more careful analysis of $\mathcal{F}'(N, k)$ is this: by definition, the last recursive call to $\mathcal{F}'(N, k-1)$ terminated with bits $k-1$ and $k-2$ being independent and having biases at least $\tilde{\varepsilon}F_k(1 - 2^{k-N-2})$ and $\tilde{\varepsilon}F_k(1 - 2^{k-N-3})$. A few lines of calculation show that if the bias of bit k was $\tilde{\varepsilon}F_k(1-y)$ (before application of (b1)), then after the application it is at least $\tilde{\varepsilon}F_k(1-cy)$ for a fixed positive constant $c < 1$. Therefore $O(N - k)$ rounds suffice to drive the bias up to $\tilde{\varepsilon}F_k(1 - 2^{k-N-1})$.

Since procedure $\mathcal{F}'(N, k)$ makes $O(N - k)$ recursive calls to $\mathcal{F}'(N, k-1)$, the overall runtime of $\mathcal{F}'(N, N)$, and hence $\mathcal{F}(N)$, is $(N!)^{O(1)} = \exp(O(N \log N))$. This establishes Theorem 1.5. ∎

Historical notes. The application of majority gates to amplify bias began at least with von Neumann's work on fault tolerance [27]. The idea was later used as part of the design for algorithmic heat engines in [22]. An experiment to demonstrate the three-bit-majority primitive was conducted in [7]; a similar experiment was conducted by Sørensen [23], for NMR imaging amplification, before NMR quantum computers had been suggested. A simpler two-bit process, also pioneered by von Neumann [26] (for the quite different purpose of extracting fair from biased coin flips), was used for cooling in [22] and then in [5]. However, since the two-bit process amplifies bias only by order $\tilde{\varepsilon}^2$ rather than by order $\tilde{\varepsilon}$, majority gates were subsequently employed in [12]. In this section we have followed that approach but use a slightly different recursive procedure $\mathcal{F}$ to achieve scaling of the bias in powers of ϕ.

9. Discussion. *Numerical estimates.* We depict a specific way of using the PPA. Consider an ion trap quantum computer in which four qubits are reserved for preparation of ancillas, all others being devoted to the main quantum algorithm (including the fault-tolerance mechanism). Of the reserved qubits, three are "computation qubits" and one is the "refrigerant." Ion trap technology is capable of placing the refrigerant in its ground state with probability 0.95 (i.e., $\tilde{\varepsilon} = \operatorname{arctanh} 0.9 \approx 1.47$). Calculation shows that application of the PPA on the quadruple for just nine cooling steps suffices to prepare one of the qubits in the ground state with probability $1 - 10^{-4}$. This is at the conservative end of the estimates of between 10^{-4} and 10^{-2} for the fault-tolerance threshold for quantum computation. Hence after every nine cooling steps the PPA can prepare an ancilla, ready to be moved by spin exchange into the main bank of qubits (in place of a "warm" qubit generated by the fault-tolerance mechanism).

Implementation objectives. It is necessary to study the sensitivity of the model to imperfections in the cooling steps as well as in the logic gates between cooling steps, in specific experimental implementations.

Experimental algorithmic cooling also has the opportunity to produce a physically meaningful result well before producing a quantum computer. A series of papers [28, 25, 6] shows that if k qubits have bias less than 2^{-2k}, then their joint state is separable. Conversely, in the ball of radius $2^{-k/2}$ there exist nonseparable states. Liquid-state NMR experiments have not, to date, produced a demonstrably nonseparable state. Achieving this goal will require some combination of an increase in the number of coherently manipulated qubits and an increase in the individual polarization of these qubits. The latter demands implementation of new cooling techniques.

In the simple model adopted in this paper we have assumed that there is only a single refrigerant qubit. One may ask how the model is affected if the number of such qubits is proportional to the number of computation qubits. (In liquid-state NMR, for example, we can expect that nuclei of various types will be present in fixed proportions.) The answer is that while some gain is likely, the fundamental limits of the model are unchanged because with a slowdown in the cooling process by a factor of $O(n)$, the same effect can be achieved by spin exchange with a single refrigerant qubit.

The present paper leaves open whether there is a simple implementation of the PPA or whether some other simply implemented algorithm can achieve the same $\varepsilon \approx 2^{-n}$ threshold.

The necessity of cooling many qubits for quantum computation. In view of the difficulty of cooling certain kinds of quantum computers, the question was posed of whether this was truly necessary [17]. Since a uniformly mixed state is unchanged by reversible (unitary) operations, computation is impossible (the statistics of the final

state do not depend on the computation steps) unless the initial mixture can be transformed into something other than the uniform mixture. Interestingly, this does not rule out the possibility of quantum-over-classical computational speedups on devices that are initialized in a highly (though not completely) mixed state. A key example was provided in [17]: the trace of a unitary operator of dimension 2^n can be computed on a device with n qubits, of which just one is strongly biased while the others are maximally mixed. (For related recent work see [20].) However, it was demonstrated in [2] that there is no way of directly simulating general quantum computers on highly mixed devices such as this. Hence computations on such devices can be accomplished, if at all, only with tailor-made algorithms. The available evidence suggests that such devices, even if noise-free, would be strictly weaker than general-purpose quantum computers, and so the suggestion in [17] is unlikely to circumvent the need for effective cooling. The necessity of using ancillas to compensate for noise buttresses this conclusion.

Summary. We have studied the fundamental limits of open-system "heat-bath" cooling, with a view to the significance of such methods for quantum computation as well as for imaging tasks limited by imperfect state preparation. We have provided a cooling (bias amplification) method and shown the following: (a) The bias it achieves is substantially higher than in previous methods, and the ground-state probability after any number of cooling steps is highest possible. (b) The number of cooling steps it requires is asymptotically close to best possible. (c) There is a sharp threshold for the heat-bath temperature, above which substantial cooling is impossible in any method, and below which it is achieved by ours.

Acknowledgments. Thanks go to R. Laflamme and J. Fernandez for helpful discussions, and to an anonymous referee for a careful reading of the manuscript.

REFERENCES

[1] D. Aharonov and M. Ben-Or, *Fault-tolerant quantum computation with constant error*, in Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, 1997, pp. 176–188.

[2] A. Ambainis, L. J. Schulman, and U. Vazirani, *Computing with highly mixed states*, in Proceedings of the Thirty-Second Annual ACM Symposium on the Theory of Computing, 2000, pp. 705–714.

[3] M. D. Barrett, B. DeMarco, T. Schaetz, V. Mayer, D. Leibfried, J. Britton, J. Chiaverini, W. M. Itano, B. Jelenkovic, J. D. Jost, C. Langer, T. Rosenband, and D. J. Wineland, *Sympathetic cooling of* $^9be^+$ *and* $^24mg^+$ *for quantum logic*, Phys. Rev. A, 68 (2003), 042302.

[4] C. M. Bowden, J. P. Dowling, and S. P. Hotaling, *Quantum computing using electron-nuclear double resonances*, in SPIE Proceedings 3076: Photonic Quantum Computing, 1997, pp. 173–182.

[5] P. O. Boykin, T. Mor, V. Roychowdhury, F. Vatan, and R. Vrijen, *Algorithmic cooling and scalable NMR quantum computers*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 3388–3393.

[6] S. L. Braunstein, C. M. Caves, R. Josza, N. Linden, S. Popescu, and R. Schack, *Separability of very noisy mixed states and implications for NMR quantum computing*, Phys. Rev. Lett., 83 (1999), pp. 1054–1057.

[7] D. E. Chang, L. M. K. Vandersypen, and M. Steffen, *NMR implementation of a building block for scalable quantum computation*, Chem. Phys. Lett., 338 (2001), pp. 337–344.

[8] J. I. Cirac and P. Zoller, *Quantum computations with cold trapped ions*, Phys. Rev. Lett., 74 (1995), pp. 4091–4094.

[9] D. G. Cory, A. F. Fahmy, and T. F. Havel, *Ensemble quantum computing by nuclear magnetic resonance spectroscopy*, Proc. Nat. Acad. Sci. USA, 94 (1997), pp. 1634–1639.

[10] D. P. DiVincenzo, *Topics in quantum computers*, in Mesoscopic Electron Transport, Kluwer, Dordrecht, 1997, pp. 657–667.
[11] D. P. DiVincenzo, *The physical implementation of quantum computation*, Fortschr. Phys., 48 (2000), pp. 771–783.
[12] J. M. Fernandez, S. Lloyd, T. Mor, and V. Roychowdhury, *Algorithmic cooling of spins: A practicable method for increasing polarization*, Internat. J. Quantum Inf., 2 (2004), pp. 461–477.
[13] N. A. Gershenfeld and I. L. Chuang, *Bulk spin-resonance quantum computation*, Science, 275 (1997), pp. 350–356.
[14] M. Iinuma, Y. Takahashi, I. Shaké, M. Oda, A. Masaike, T. Yabuzaki, and H. M. Shimizu, *High proton polarization by microwave-induced optical nuclear polarization at* 77 *K*, Phys. Rev. Lett., 84 (2000), pp. 171–174.
[15] B. E. King, C. S. Wood, C. J. Myatt, Q. A. Turchette, D. Leibfried, W. M. Itano, C. Monroe, and D. J. Wineland, *Cooling the collective motion of trapped ions to initialize a quantum register*, Phys. Rev. Lett., 81 (1998), pp. 1525–1528.
[16] E. Knill, *Fault-Tolerant Postselected Quantum Computation: Threshold Analysis*, http://arxiv.org/abs/quant-ph/0404104 (2004).
[17] E. Knill and R. Laflamme, *On the power of one bit of quantum information*, Phys. Rev. Lett., 81 (1998), pp. 5672–5675.
[18] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and its Applications*, Academic Press, New York, London, 1979.
[19] C. Monroe, D. M. Meekhof, B. E. King, W. M. Itano, and D. J. Wineland, *Demonstration of a fundamental quantum logic gate*, Phys. Rev. Lett., 75 (1995), pp. 4714–4717.
[20] D. Poulin, R. Blume-Kohout, R. Laflamme, and H. Ollivier, *Exponential speed-up with a single bit of quantum information: Measuring the average fidelity decay*, Phys. Rev. Lett., 92 (2004), 177906.
[21] L. J. Schulman, T. Mor, and Y. Weinstein, *Physical limits of heat-bath algorithmic cooling*, Phys. Rev. Lett., 94 (2005), 120501.
[22] L. J. Schulman and U. Vazirani, *Molecular scale heat engines and scalable quantum computation*, in Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing, 1999, pp. 322–329.
[23] O. W. Sørensen, *Polarization transfer experiments in high-resolution NMR spectroscopy*, Prog. NMR Spec., 21 (1989), pp. 504–569.
[24] A. S. Verhulst, O. Liivak, M. H. Sherwood, H.-M. Vieth, and I. L. Chuang, *Non-thermal nuclear magnetic resonance quantum computing using hyperpolarized xenon*, Appl. Phys. Lett., 79 (2001), pp. 2480–2482.
[25] G. Vidal and R. Tarrach, *Robustness of entanglement*, Phys. Rev. A, 59 (1999), pp. 141–155.
[26] J. von Neumann, *Various techniques used in connection with random digits*, in The Monte Carlo Method, Nat. Bur. Standards Appl. Math. Ser. 12, U.S. Government Printing Office, Washington, D.C., 1951, pp. 36–38.
[27] J. von Neumann, *Probabilistic logics and the synthesis of reliable organisms from unreliable components*, in Automata Studies, C. E. Shannon and J. McCarthy, eds., Princeton University Press, Princeton, NJ, 1956, pp. 43–98.
[28] K. Zyczkowski, P. Horodecki, A. Sanpera, and M. Lewenstein, *Volume of the set of separable states*, Phys. Rev. A, 58 (1998), pp. 883–892.